**Power and logarithmic functions**

math.eXp(X)

Return e raised to the power X, where e = 2.718281… is the base of natural logarithms. This is usually more accurate than math.e \*\* X or pow(math.e, X).

math.eXpm1(X)

Return e raised to the power X, minus 1. Here e is the base of natural logarithms. For small floats X, the subtraction in eXp(X) - 1 can result in a significant loss of precision; the eXpm1() function provides a way to compute this quantity to full precision:

math.log(X[, base])

With one argument, return the natural logarithm of X (to base e).

With two arguments, return the logarithm of X to the given base, calculated as log(X)/log(base).

math.log1p(X)

Return the natural logarithm of 1+X (base e). The result is calculated in a way which is accurate for X near zero.

math.log2(X)

Return the base-2 logarithm of X. This is usually more accurate than log(X, 2).

.

See also int.bit\_length() returns the number of bits necessary to represent an integer in binary, eXcluding the sign and leading zeros.

math.log10(X)

Return the base-10 logarithm of X. This is usually more accurate than log(X, 10).

math.pow(X, y)

Return X raised to the power y. EXceptional cases follow AnneX ‘F’ of the C99 standard as far as possible. In particular, pow(1.0, X) and pow(X, 0.0) always return 1.0, even when X is a zero or a NaN. If both X and y are finite, X is negative, and y is not an integer then pow(X, y) is undefined, and raises ValueError.

Unlike the built-in \*\* operator, math.pow() converts both its arguments to type float. Use \*\* or the built-in pow() function for computing eXact integer powers.

math.sqrt(X)

Return the square root of X.